2021 IBM Systems

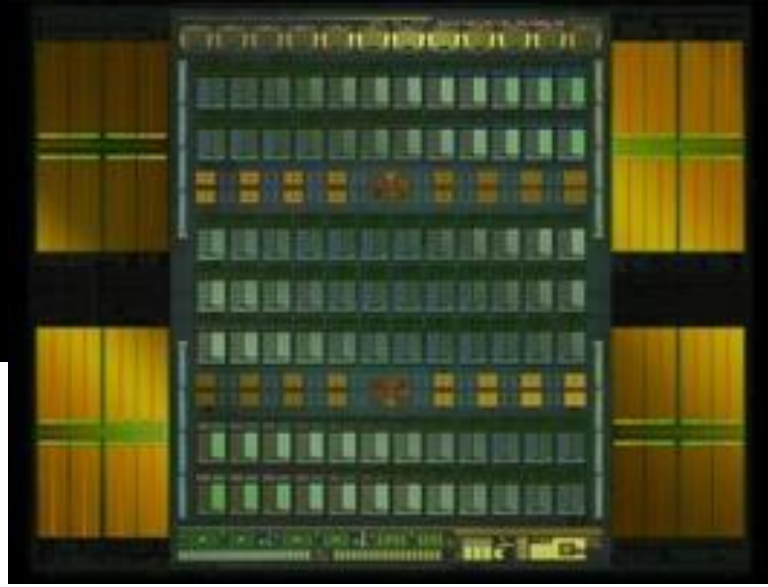
Jean-Armand Broyelle

ML/DL Cognitive Systems Technical Leader
IBM Garage for Systems - Montpellier France

Improve DeepLearning Analysis of Large Images with IBM WML-CE

IBM

Deep learning NN Training is memory constrained



- GPUs have limited memory
- Neural networks are growing deeper and wider
- Amount and size of data to process is always growing to improve AI solutions accuracy

Dimension problem with Medical Images

How much can a NVIDA Tesla V100 (32Gb) take in ?

- ResNet-101, batch size=32, it can take in images of 512*512*3 → 750kB * 32 → 24 MB
- ResNet-101, batch size=1, it can take in image of 3880*3880*3 → 43 MB

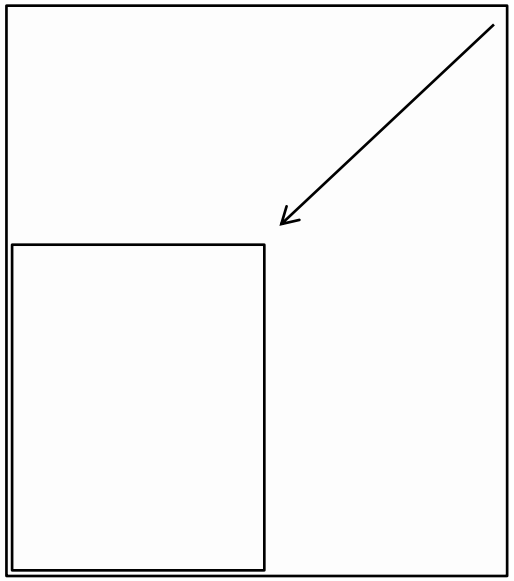
- 3D ResNet-101, batch size=32, it can take in images of 92*92*42*1 → 339kB *32 → 10 MB
- 3D ResNet-101, batch size=1, it can take in image of 577*577*42* → 13MB

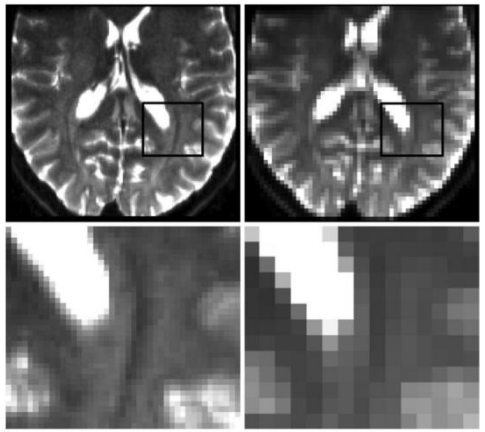
But Typical Resolution of Medical Image are LARGE

- Chest radiograph : 4000*5000 uint16 → 305MB
- Computed tomography : 512*512*50 uint16 → 200MB
- Low-dose lung CT: 512*512*500 uint16 → 2GB
- Digital Whole Slide Image : 100,000*50,000*3 uint8 → 111GB

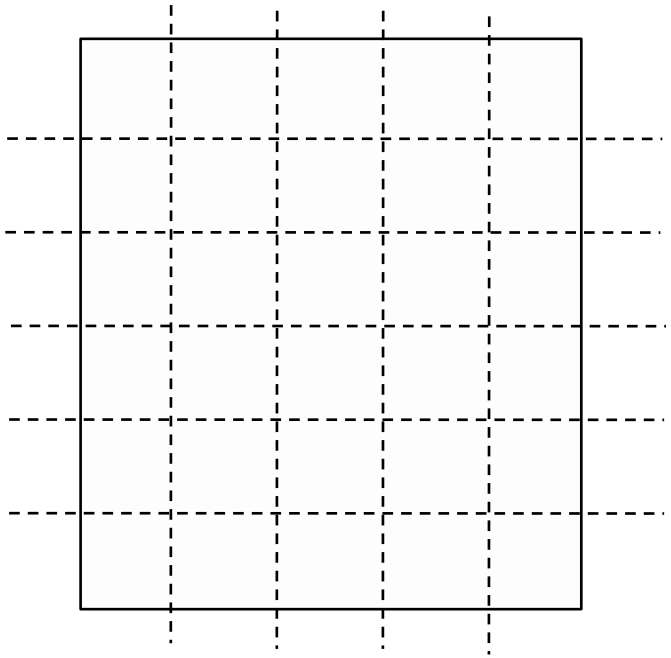
Current approaches to deal with NN models size constraints with Large images

Down-sample





Patch/tiles-based methods



→ Information loss
→ Low detection of tiny detailled objects

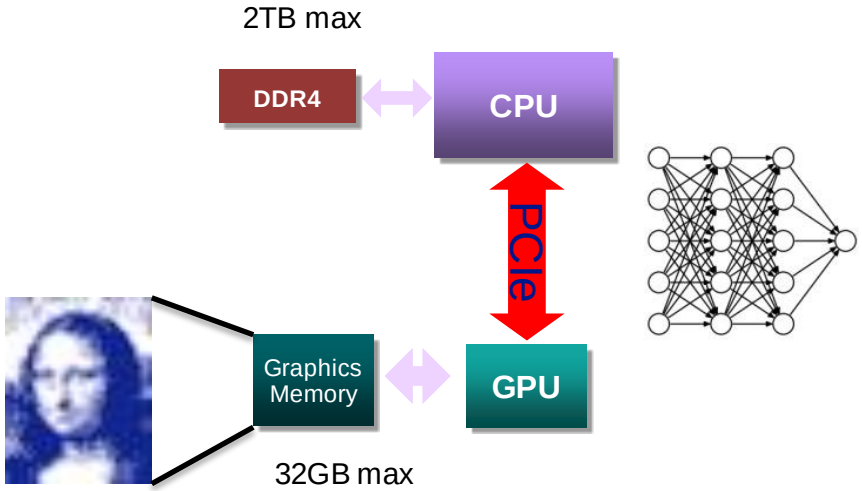
→ Low detection of objects with different scaling
→ Complex post-processing

WML-CE introduce Large Model Support

Train larger images, more complex models

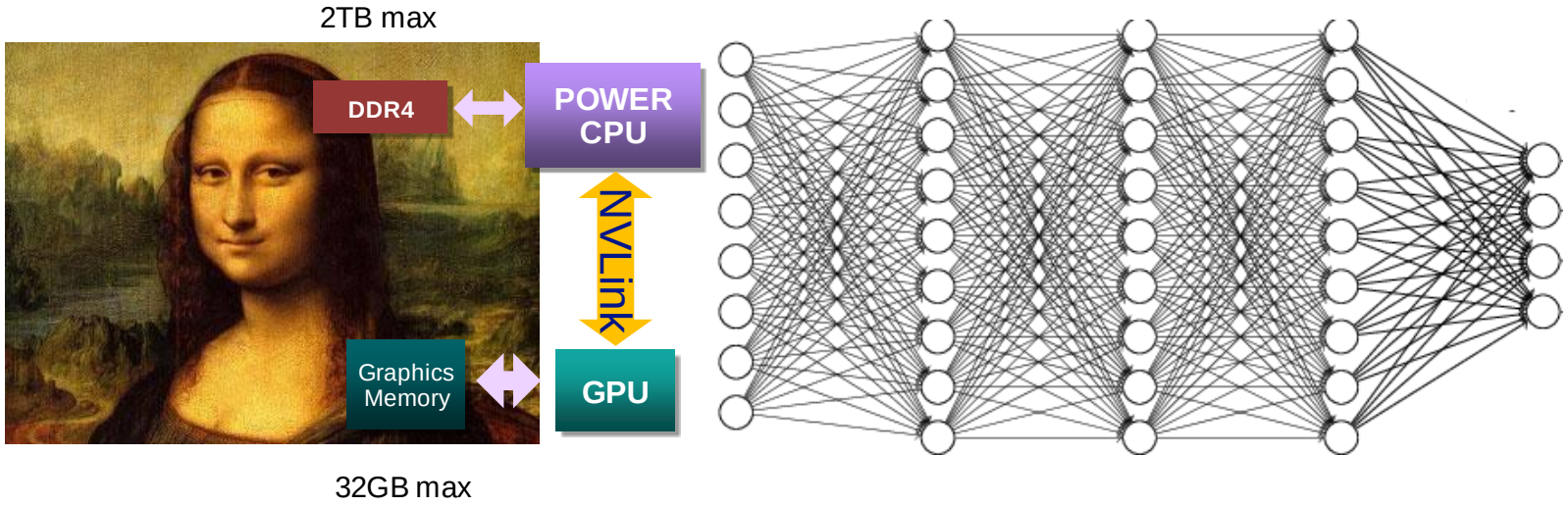
Traditional Model Support

Limited memory on GPU forces trade-off in model size / data resolution



Large Model Support

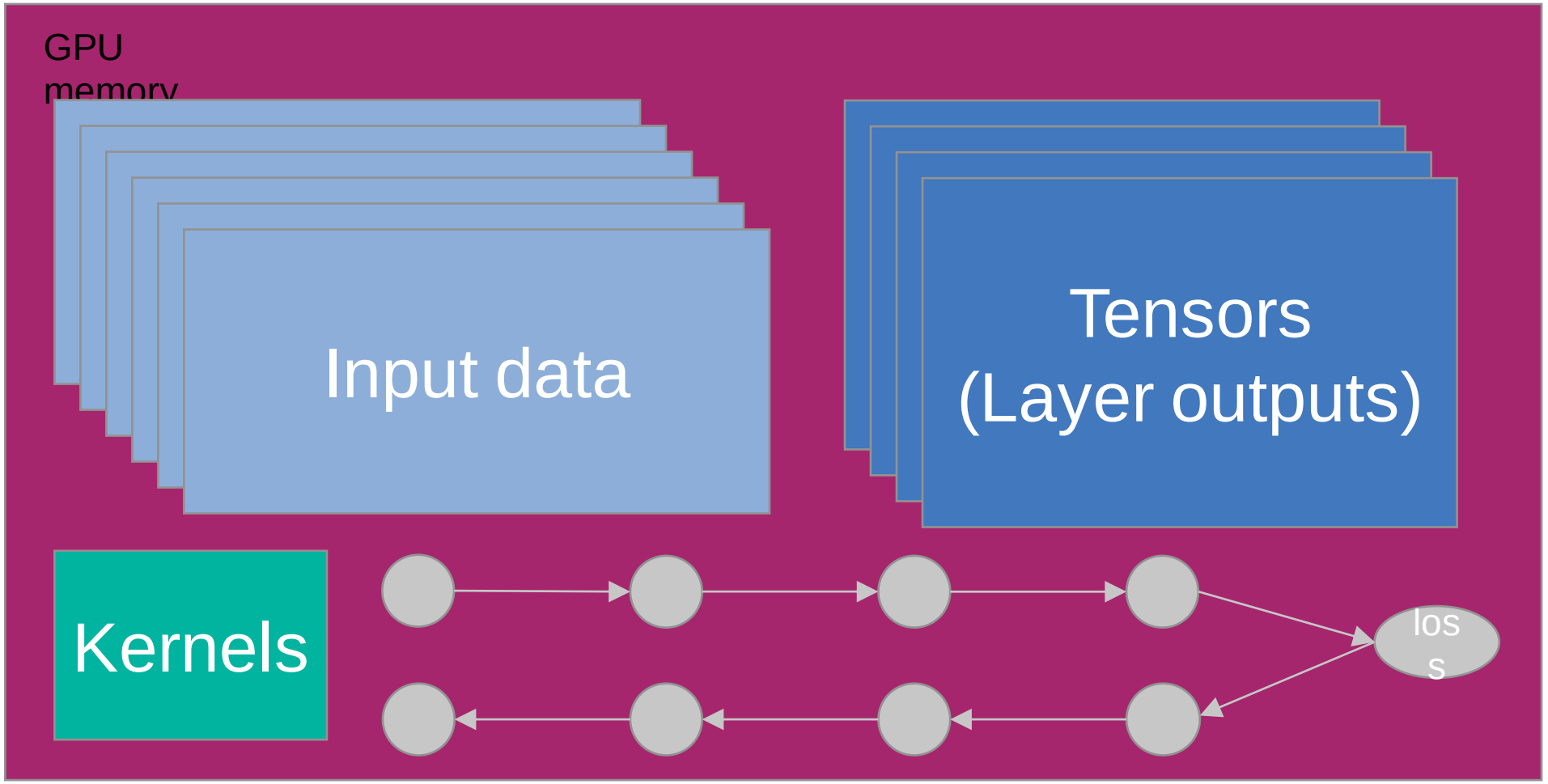
Use system memory and GPU to support more complex and higher resolution data



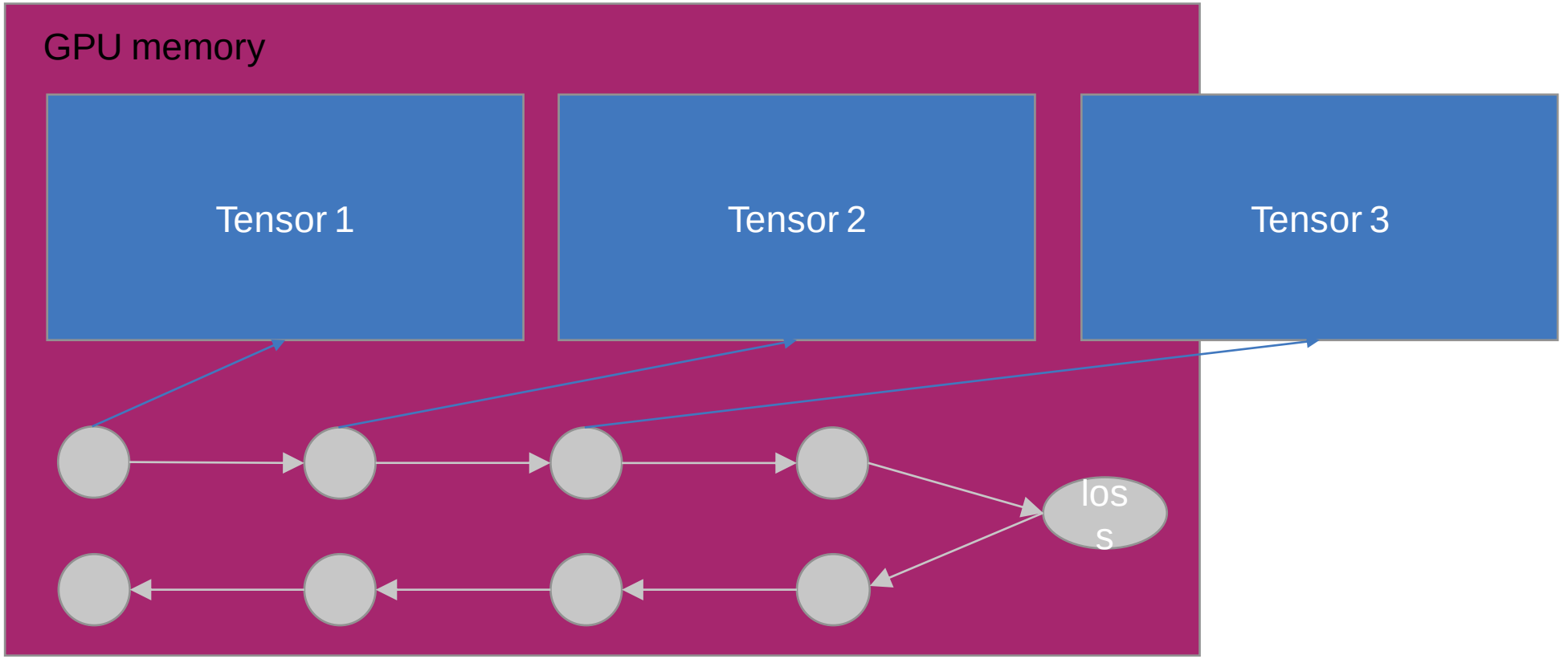
Leveraging NVLink and CPU-GPU memory coherence enables larger and more complex models

Improves model accuracy with more images and higher resolution images

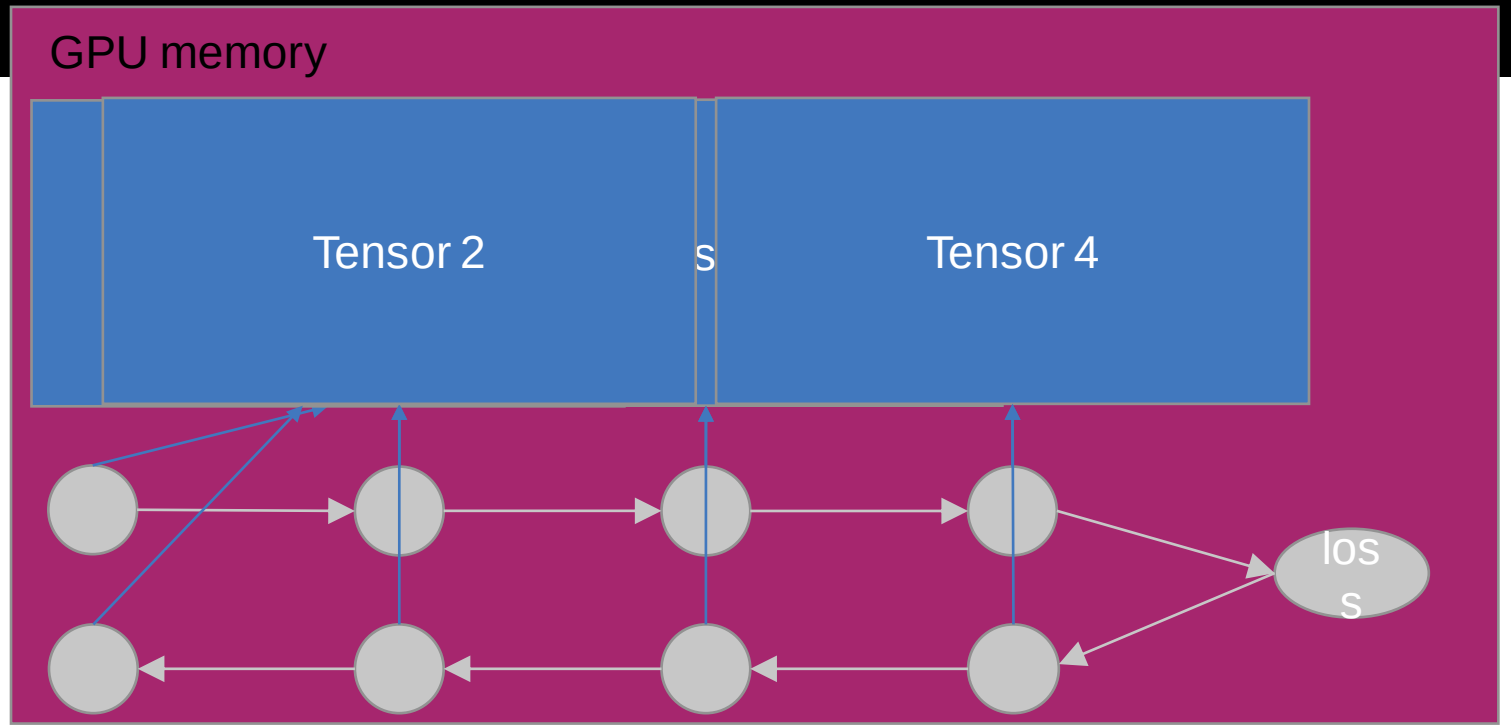
Gpu memory usage

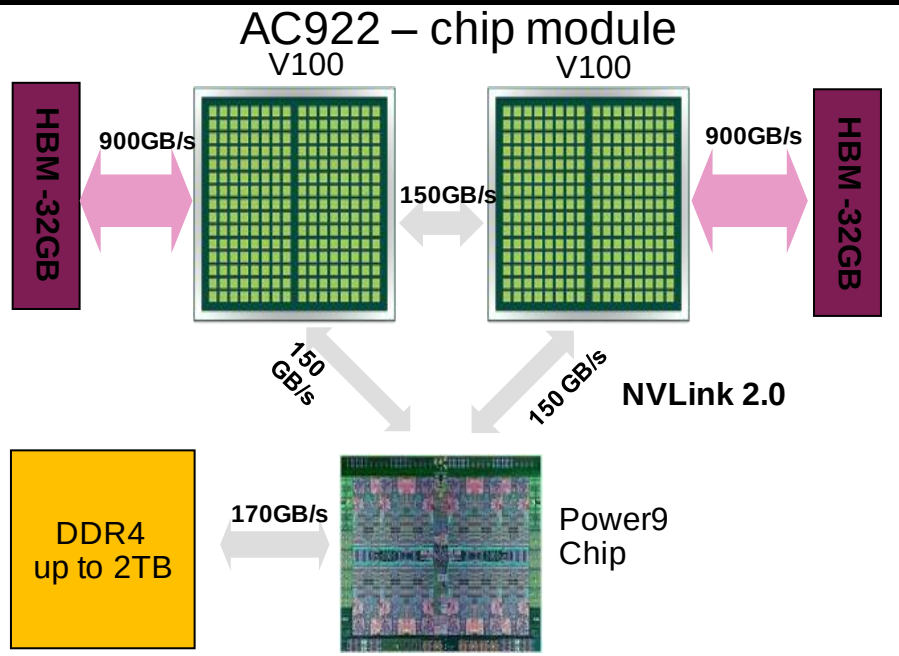


Model training in gpu memory



Model training with tensor swapping





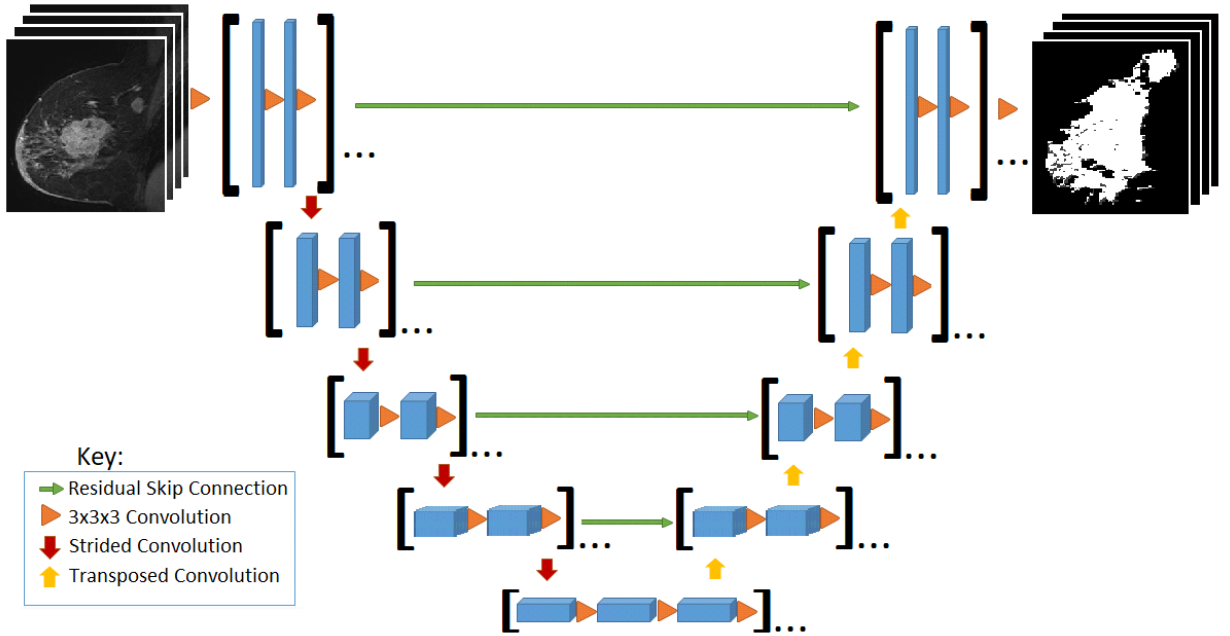
System memory

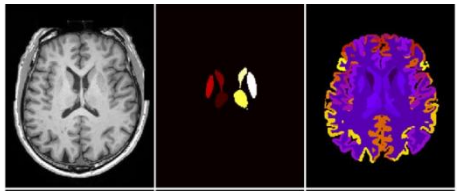
What’s possible with Large model support?

- GoogleNet: 10x image resolution
- Resnet50: 10x image resolution
- Resnet152: 10x image resolution
- 3D-Unet brain MRIs: 5x image resolution
- DeepLabV3 2D : 10 x image resolution

1st use-case : 3D-UNet image segmentation 3D Dicom images

- 3D-Unet generally has high memory usage requirements
  - 3D Dicom image size > 100 MB
- International Multimodal Brain Tumor Segmentation Challenge (BraTS)
- Existing Keras model with TensorFlow backend
- Experiment with TensorFlow Large Model Support (TFLMS)

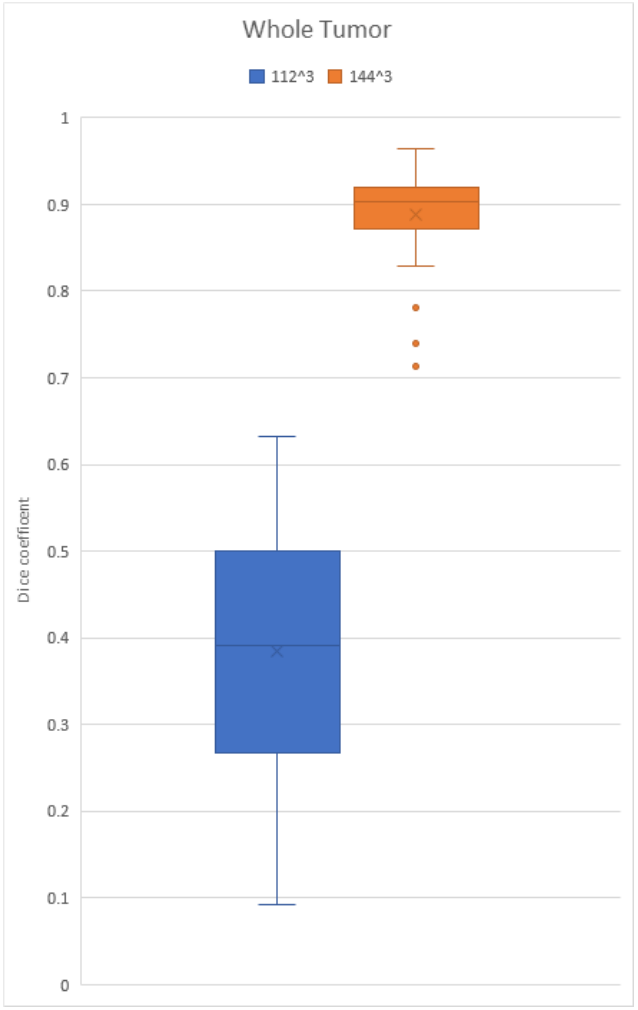


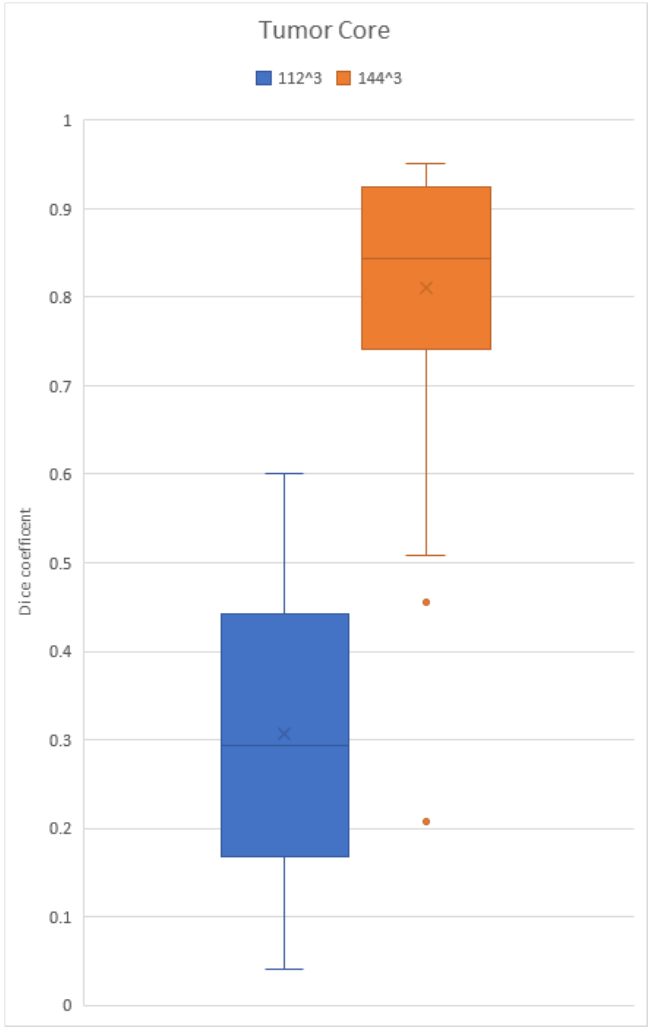


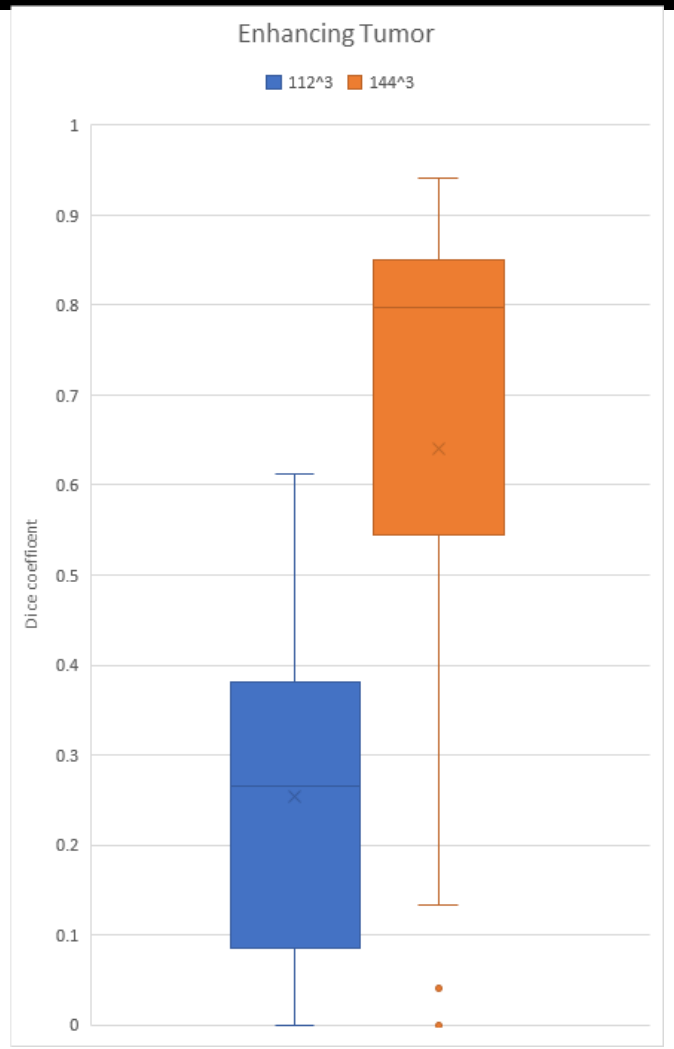
Perelman School of Medicine University of Pennsylvania

Effect of 2x resolution on dice coefficients

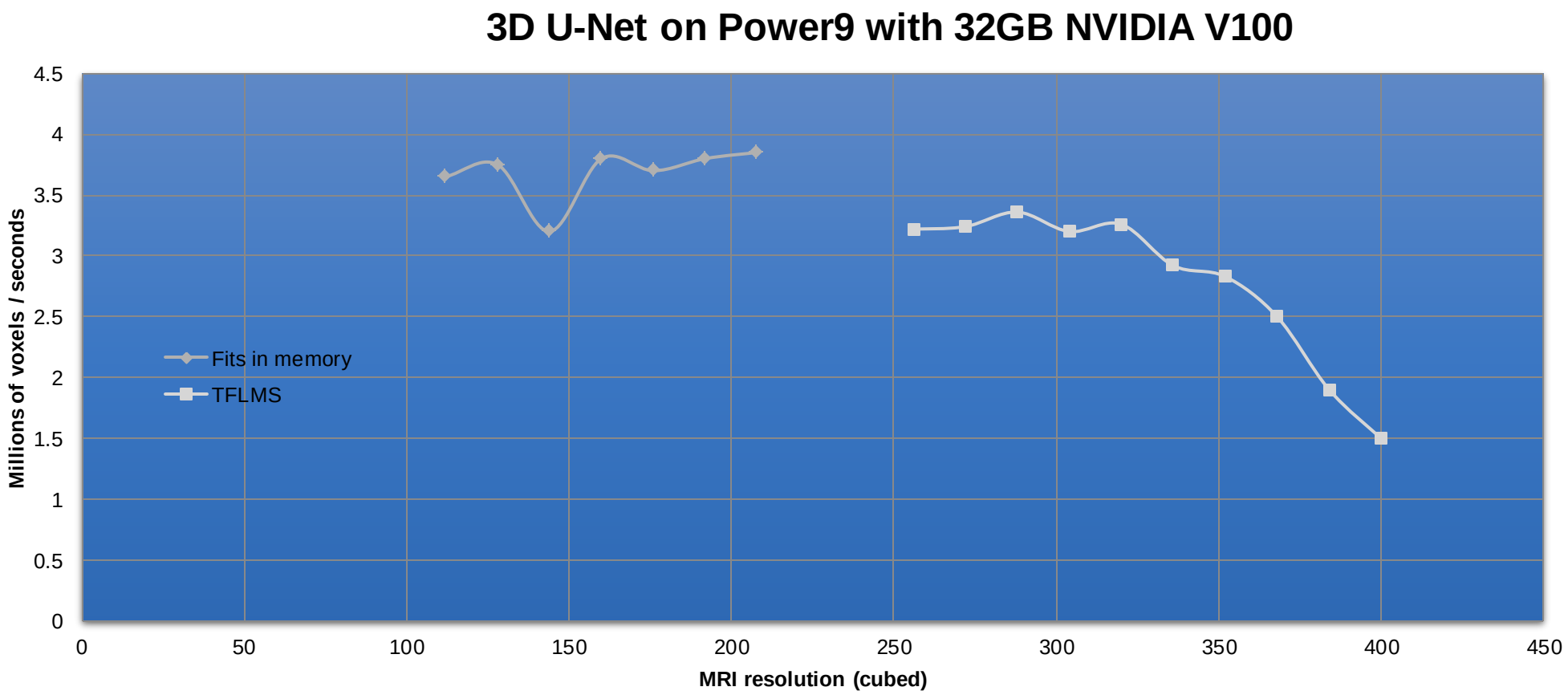
(higher is better)







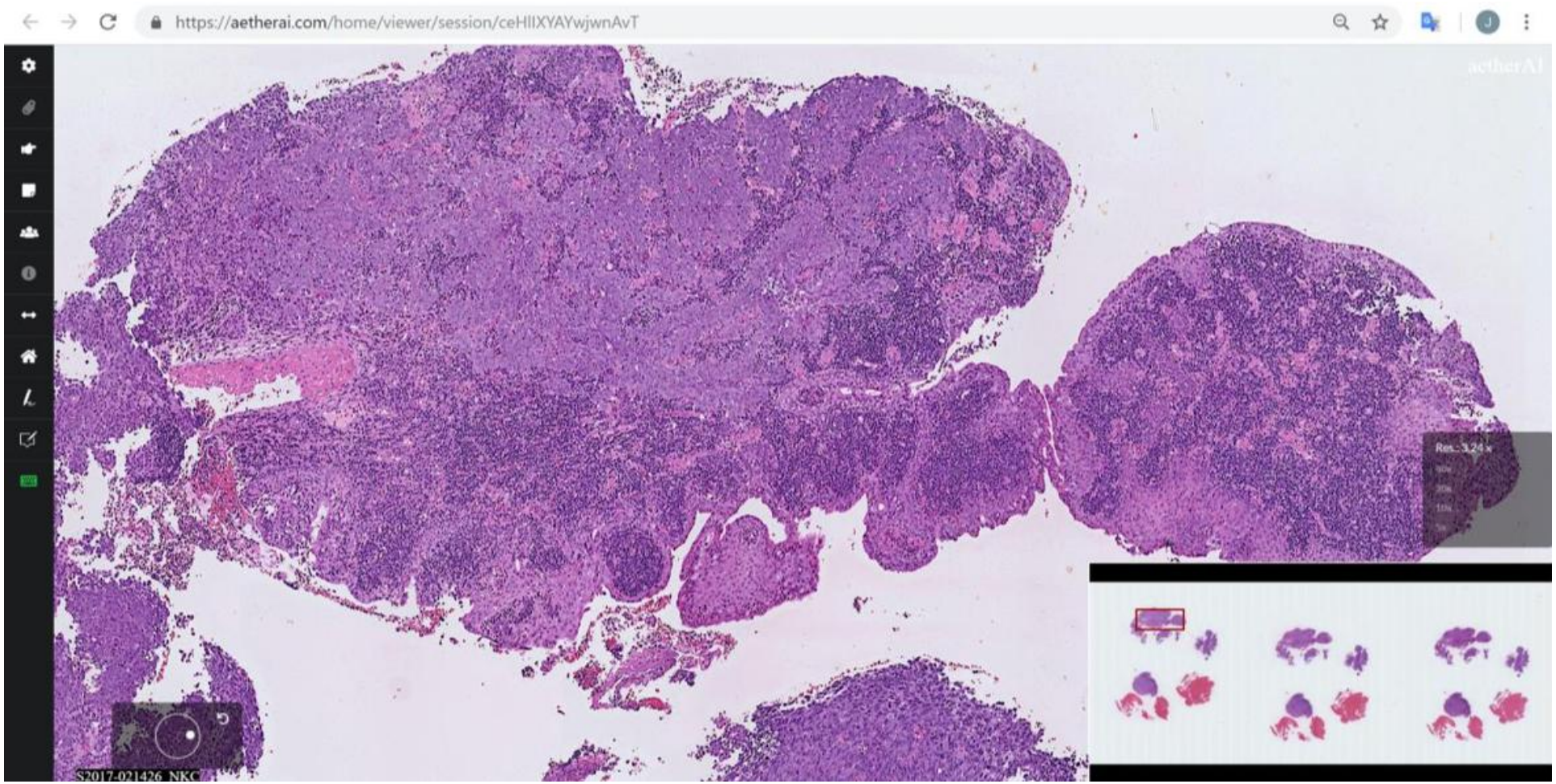
Overhead of Large Model Support with NVLink 2.0



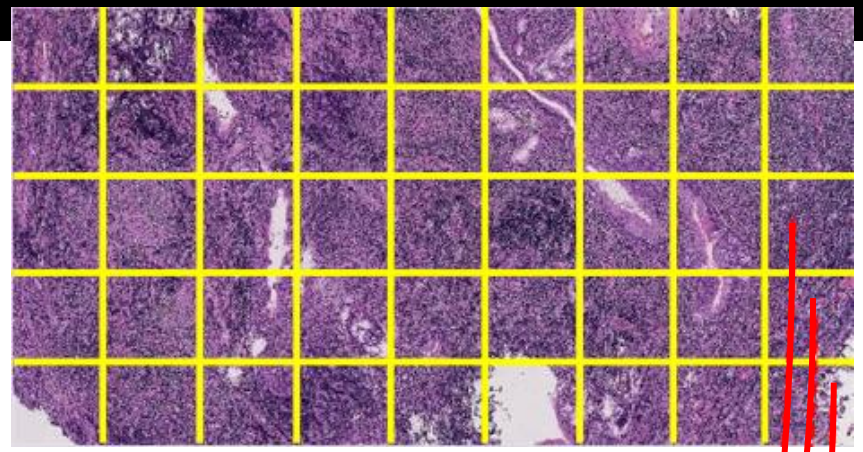
Measured with TFLMS v2.0.0 on TensorFlow 1.13, CUDA 10.1, cuDNN 7.5

2nd) Digital Whole Slide Image (WSI) ResNet Classification Experiment

- Generated by slide scanner
- Resolution can be up to 200,000 * 100,000 pixels (20 Billion) → 447 GB



Curent approach: Two-Level AI Model for Cancer Detection on Whole Slide Image

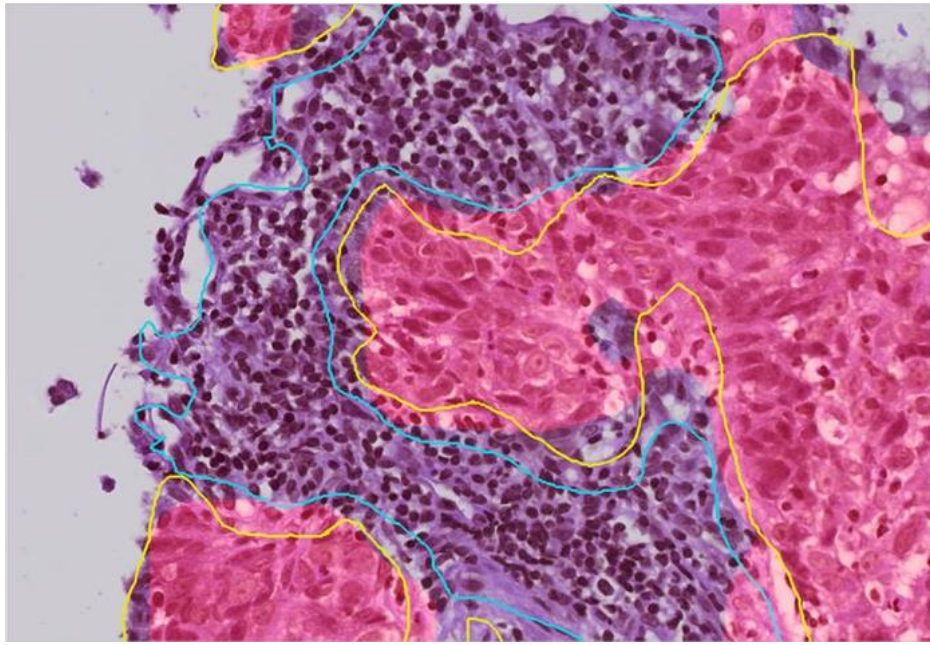


Divide WSI
into patches

Patch-level model (>10M Patches)
Background, Benign, Cancer
Classification accuracy : 98%

Slide-level model
260 Training, 100 Testing
Classification Accuracy : 97%

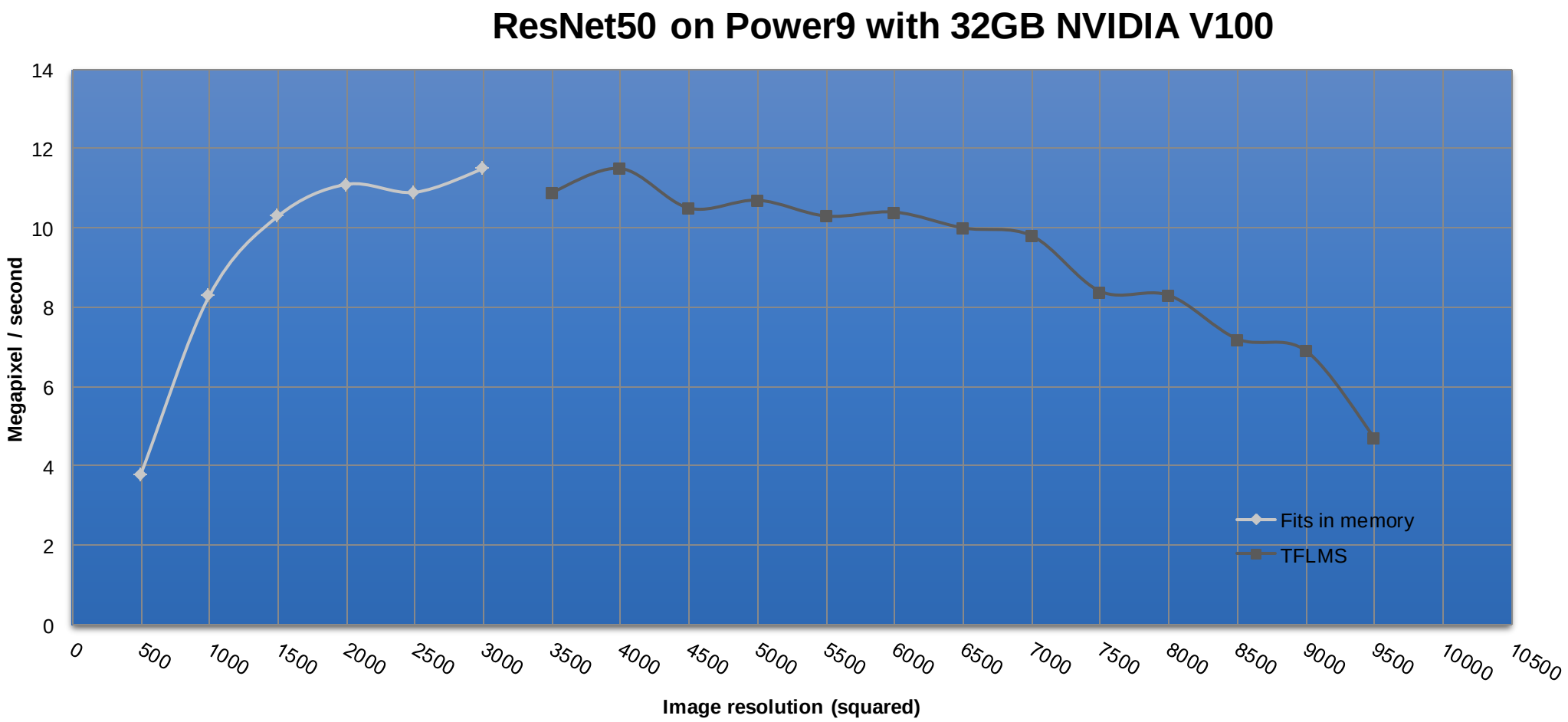
Benign or Nasopharyngeal
Cancer ?



Ground Truth : Cancer, Normal Tissue
Shadowed area : Cancer predicted by AI

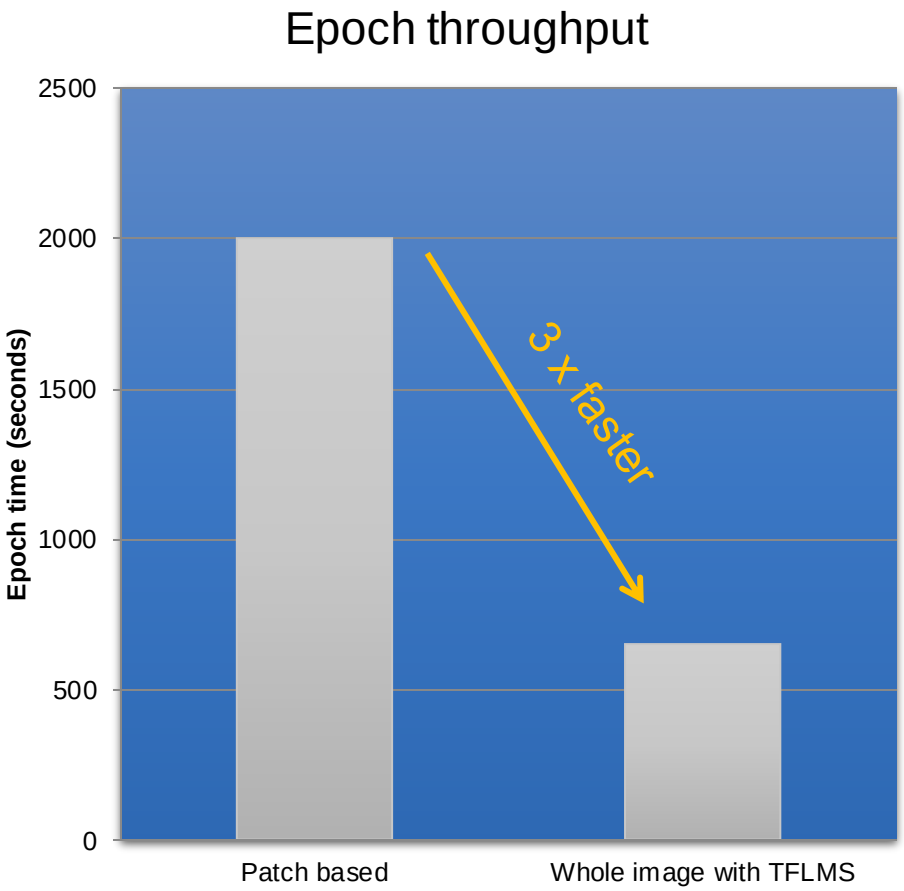
Overhead of Large Model Support with NVLink 2.0

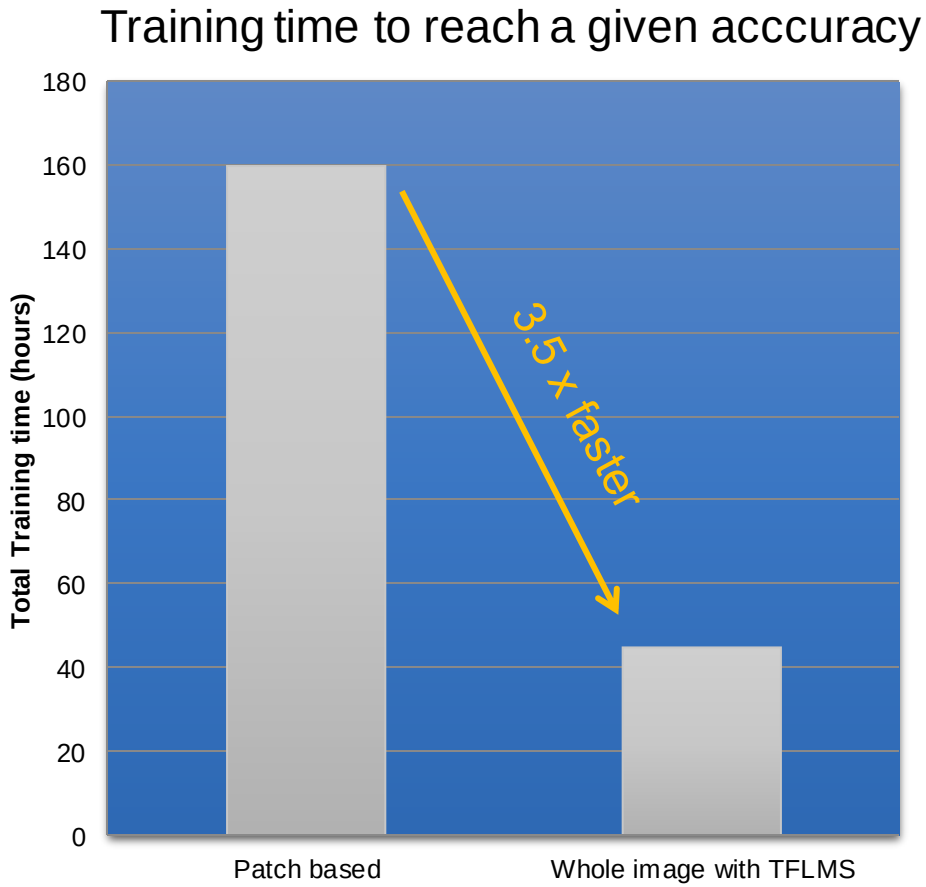
ResNet50 and LMS on AC922 V100 32GB



Measured with TFLMS v2.0.0 on TensorFlow 1.13, CUDA 10.1, cuDNN 7.5

Patches versus whole image



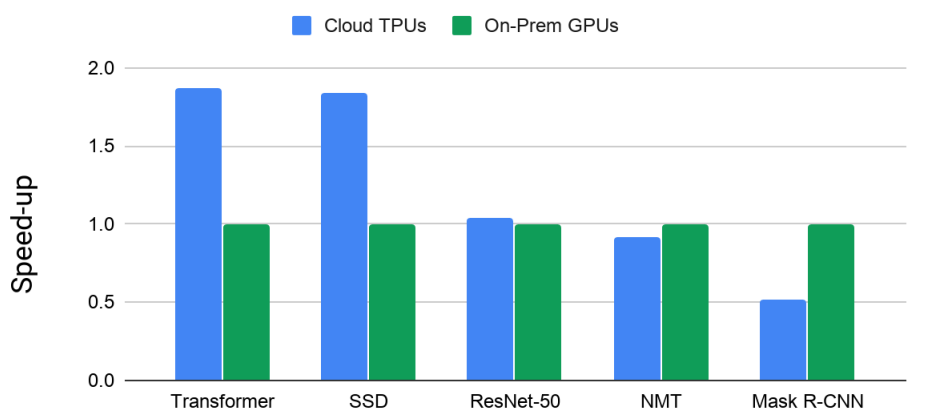


https://arxiv.org/abs/1812.07816

3td) NLP : train Bert Large with LMS

Requirements and limitations

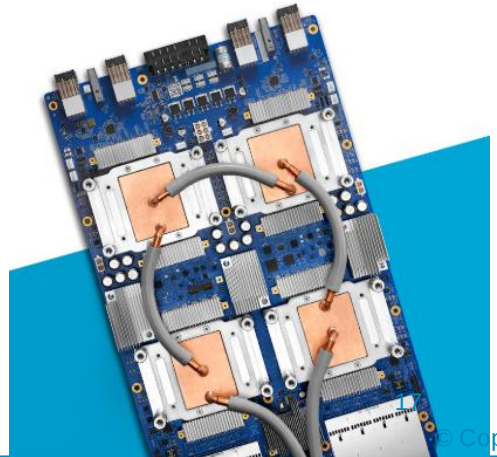
- **High compute**
  - Training made on 16 Google Cloud TPUs (64 chips)
- **High memory**
  - Requires 100+ GB for training [TODO verify / graph ?]



MLPerf 0.6 Benchmark

Cloud TPU v3 vs. Nvidia DGX-2h with GPUs

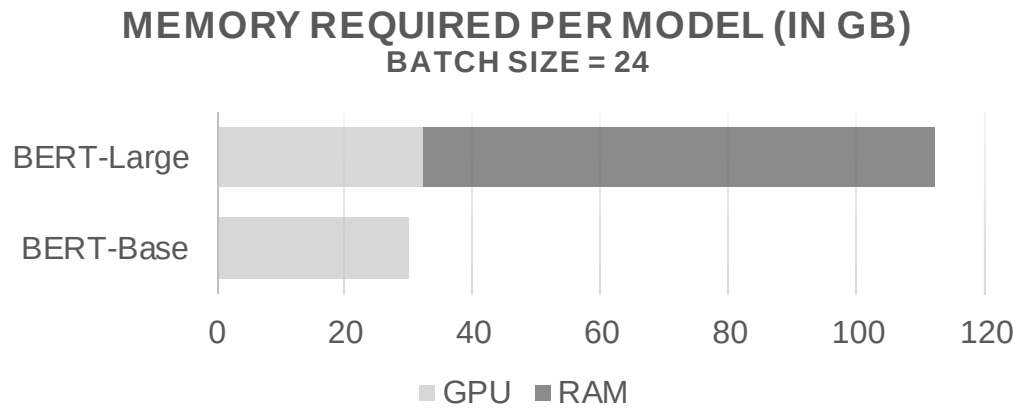
Impossible to train **$BERT_{Large}$** on best **GPUs** (32GB) without reducing network and parameters (batch size = 1), which **harms the final accuracy.**



Bert Large with LMS : Results

Goals:

- Prove that LMS allows to train or finetune BERT-Large on a single GPU



- Reproduce accuracy achieved by Google AI (on Cloud TPUs)

IBM Power AC922

Nvidia Tesla V100 (32GB) GPU

IBM Large Model Support

4th) Automotive Driving : PSPNET training against HD images with LMS

Large images allow to see far & anticipate.

IBM PowerAI

Needed memory More than 25GB	Training duration 10 days of training	Dataset : CityScapes Model: PSPNET
Large Model Support WML-ce LMS to use RAM as an extension of GPU memory	Distributed Deep Learning WML-ce DDL to use multiple GPUs on multiple nodes for parallel computation	With LMS, the accuracy ~75% of intersection-over-union class accuracy compared to ~73% with lower resolution images. DDL reduce training time by 40%.

Tensorflow Large Model implementation overview

Keras API

```
from tensorflow_large_model_support import LMS
Lms = LMS()
Lms.batch_size = 1
…
model.fit_generator(generator=training_gen,
                    callbacks=[lms])
```

TF Estimator API

```
from tensorflow_large_model_support import LMS
lms = LMS()
…
mnist_classifier.train(input_fn=train_input_fn, steps=20000,
                       hooks=[logging_hook, lms])
```

PyTorch LMS - Usage and Tuning

Extends PyTorch's torch.cuda package to provide the following control and tuning interfaces:

```
torch.cuda.set_enabled_lms(enable)
```

Enable/disable Large Model Support.
Parameters: **enable** (bool): desired LMS setting.

```
torch.cuda.set_limit_lms(limit)
```

Sets the allocation limit (in bytes) for LMS.
Parameters: **limit** (int): soft limit on GPU memory allocated for tensors.

```
torch.cuda.set_size_lms(size)
```

Sets the minimum size (in bytes) for LMS.
Parameters: **size** (int): any tensor smaller than this value is exempt from LMS reuse and persists in GPU memory.

PyTorch Large Model Support - Consideration

- Ensure active tensors are resident in device memory during computations. Maintain set of inactive tensors eligible for eviction.
- Enhance GPU allocator to allow eviction of inactive tensors as an alternative to new device memory allocation.
- All tensors, including weights/biases, are eligible for swapping.
- No graph to analyze, but PyTorch efficiently manages tensor lifespans.
- Implementing heuristics which are informed by prior iterations is a challenge:
  - (Most) tensors do not persist across iterations
  - Conditional control flow allows iterations to differ from one to the next
- Next memory limitations to consider :
  - RAM capacity (to avoid system swapping)
  - CudNN : single tensor indexing is based on INT32 → the size of one single tensor can't exceed 2Giga elements of tensor 'datatype'

PyTorch LMS - Statistics

Extends PyTorch's torch.cuda package to provide the following statistics interfaces:

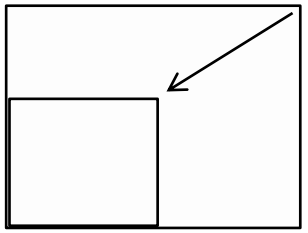
```
# Memory active:
    torch.cuda.memory_active()
    torch.cuda.max_memory_active()
    torch.cuda.reset_memory_active()

# Memory swapped:
    torch.cuda.memory_reclaimed()
    torch.cuda.reset_memory_reclaimed()
```

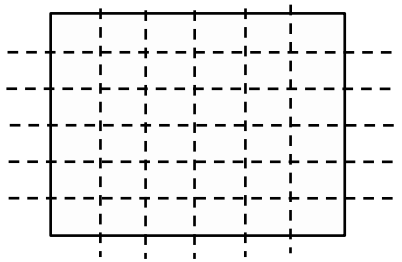
Conclusion

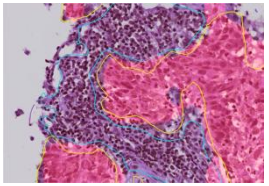
- Dimension problem with large image for AI.

  → Large Training images provide better models but hit GPU limit on traditionnal h/w

- Classical ways to overcome dimension problems could limit model performances.







- **WML-CE Large Model Support** is a unique differentiator that address this issue.
  - Native input image resolution / Reduce tiles number → Higher Accuracy
  - Increase batch-size / Less patches / Low Overhead → Lower Training Time (model dependant)
  - Simple to implement / Reduce workflow complexity → Improve Team productivity

More informations

- TensorFlow materiels
  - https://developer.ibm.com/linuxonpower/2021/06/11/tensorflow-large-model-support-resources/
  - TensorFlow Large Model Support Research Paper : https://arxiv.org/pdf/1807.02037.pdf
  - TensorFlow Large Model Support Case Study: https://developer.ibm.com/linuxonpower/2018/07/27/tensorflow-large-model-support-case-study-3d-image-segmentation/
- IBM AC922 with NVLink 2.0 connections between CPU and GPU : https://www.ibm.com/us-en/marketplace/power-systems-ac922
- PyTorch Upstream contribution in progress
  - Code: https://github.com/mtbrandy/pytorch/tree/v1.2.0-LMS
  - Wiki: https://github.com/mtbrandy/pytorch/wiki/Large-Model-Support

Notices and disclaimers

- © 2018 International Business Machines Corporation. No part of this document may be reproduced or transmitted in any form without written permission from IBM.
- **U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.**
- Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **This document is distributed "as is" without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.** IBM products and services are warranted per the terms and conditions of the agreements under which they are provided.
- IBM products are manufactured from new parts or new and used parts.
  In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."
- **Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.**
- Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those
- customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.
- References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.
- Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.
- It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

Notices and disclaimers continued

- Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products about this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**
- The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.
- IBM, the IBM logo, ibm.com and [names of other referenced IBM products and services used in the presentation] are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.
- .